SPC

Smart Peripheral Controller

Neo Stepper Motor 1.2A

Table Of Contents

1. INTRODUCTION

Smart Peripheral Controller / SPC NEO STEPPER MOTOR 1.2A is a stepper motor controller module which is compact, reliable, and compatible for robotic applications. This module can be used to control the direction and speed of 1 stepper motor using full-step or half-step. This module is equipped with UART TTL and I^2C interfaces so that it can easily be connected with other systems.

1.1. SPC NEO STEPPER MOTOR 1.2A SPECIFICATION

SPC NEO STEPPER MOTOR 1.2A specification is as follows:

- The module requires 4.8 – 5.4 VDC power supply.
- The motor requires 2.5 – 13.5 VDC power supply.
- Uses a TB6612FNG motor driver IC.
- The driver's maximum continuous current is 1.2 A.
- Can be used for unipolar or bipolar stepper motors.
- Input/Output pins are compatible with TTL and CMOS voltage level.
- Equipped with UART TTL and I^2C interface.
- Using I^2C, SPC NEO STEPPER MOTOR 1.2A can be cascaded up to 8 modules.
- Module uses a 16 pins 600mil DIP connector configuration which is easier to connect directly to project board or PCB matrix.

1.2. SUGGESTED SYSTEM

Suggested system for SPC NEO STEPPER MOTOR 1.2A is as follows:

Hardware:

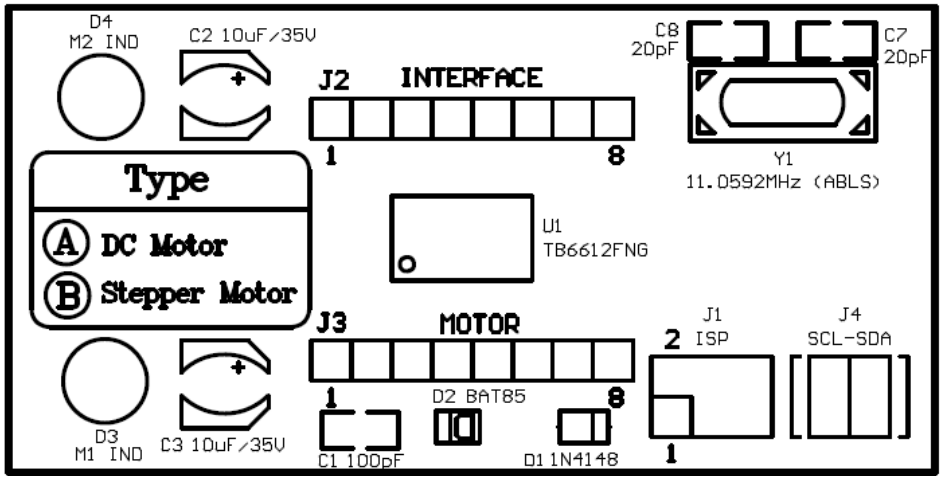
- PC™ AT™ Pentium® IBM™ Compatible with USB port.
- DT-AVR Low Cost Series.
- DVD-ROM Drive and Hard disk.

Software:

- Windows® XP Operating System.
- CodeVisionAVR©.
- program CD/DVD contents:
  Folder contoh_i2c, folder contoh_uart, TB6612FNG.pdf, dan SPC Neo Stepper Motor 1.2A Manual.pdf.

2. SPC NEO STEPPER MOTOR 1.2A HARDWARE

2.1. SPC NEO STEPPER MOTOR 1.2A COMPONENT LAYOUT



2.2. CONNECTORS AND JUMPERS CONFIGURATIONS

INTERFACE PORT (J2) connector functions as a connector for module power supply input, UART TTL, and I^2C interface.

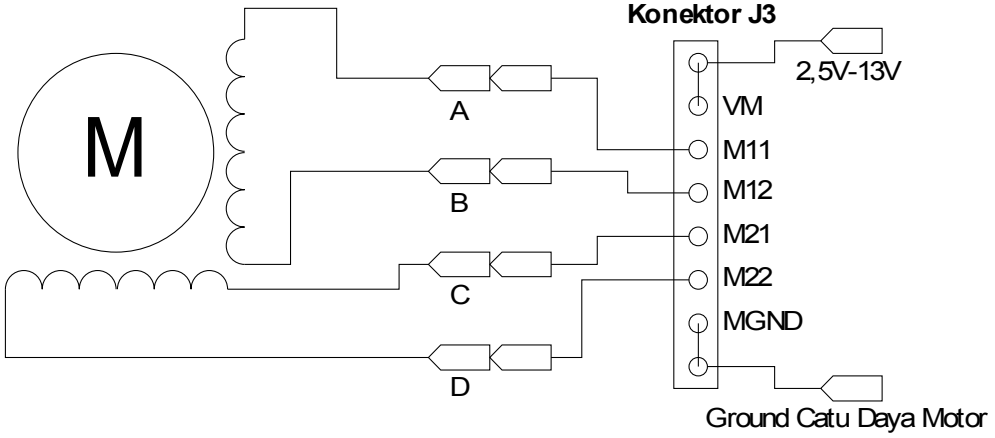
Pin	Name	Function
1,2	VIN	Connected to power supply (4.8 – 5.4 Volts)
3	SCL	I^2C-bus clock input
4	SDA	I^2C-bus data input / output
5	RX TTL	TTL serial level input to SPC module
6	TX TTL	TTL serial level output from SPC module
7,8	PGND	Ground reference for SPC module power supply

MOTOR (J3) connector functions as a connector for motor power supply input and motors.

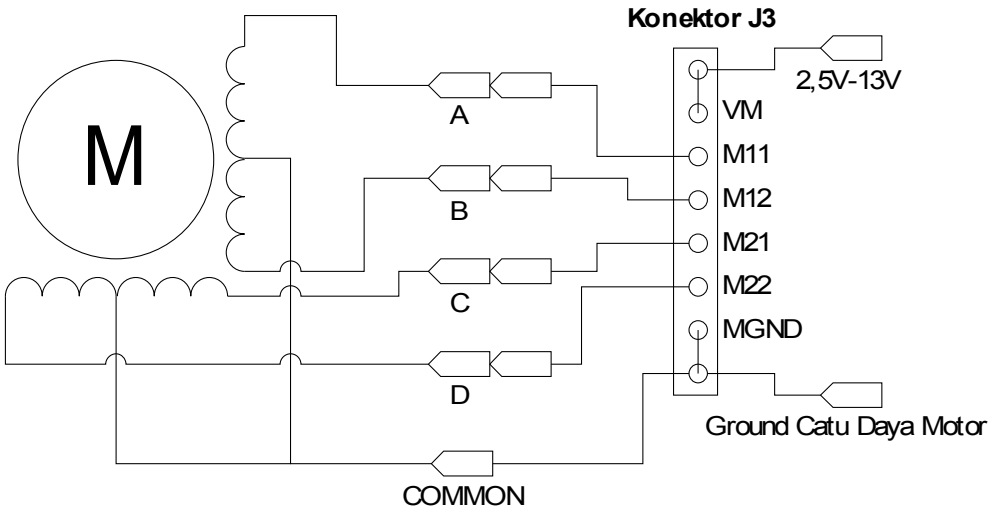
Pin	Name	Function
1,2	VM	Connected to motor power supply (2.5 – 13.5 Volts)
3	M11	1st Output from H-Bridge **M1** pair
4	M12	2nd Output from H-Bridge **M1** pair
5	M21	1st Output from H-Bridge **M2** pair
6	M22	2nd Output from H-Bridge **M2** pair
7,8	MGND	Ground reference for motor power supply

Pay attention to the type of stepper motor connected to SPC NEO STEPPER MOTOR 1.2A because each type has its own connection. SPC NEO STEPPER MOTOR 1.2A can be utilized for 3 types of stepper motor: Bipolar, 5 cables Unipolar, and 6 cables Unipolar. The following are the connection examples for each stepper motor type:

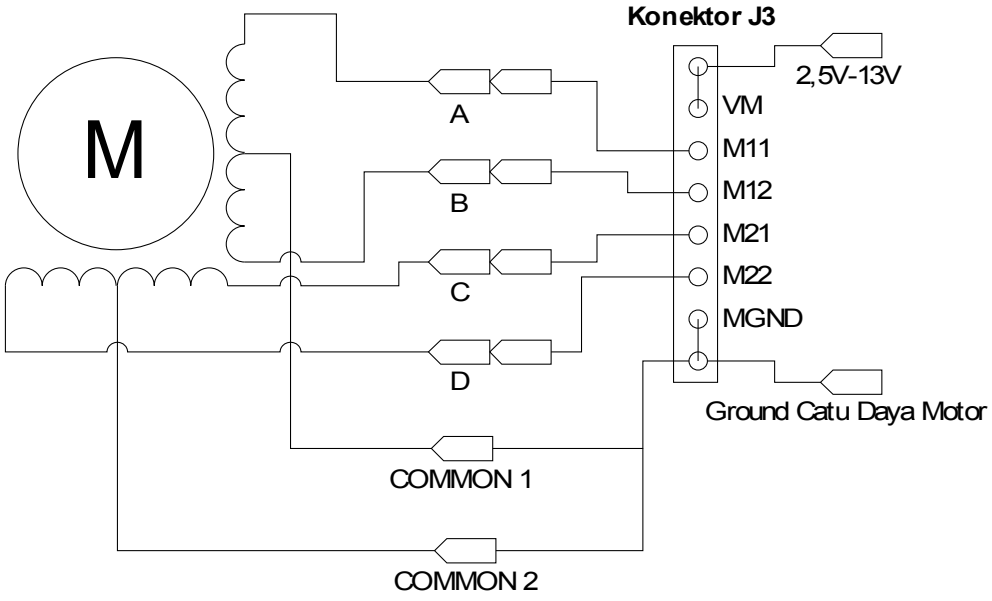
Bipolar



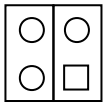
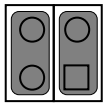
Unipolar 5 kabel



Unipolar 6 kabel



SCL-SDA (J4) jumpers are used to activate pull-up resistors for SDA and SCL on I^2C interface.

Jumper SCL-SDA J4	**Function**
 SCL SDA	Pull-up inactive (jumpers disconnected)
 SCL SDA	Pull up active (jumpers connected)

Important!
If more than one module is connected to I²C-bus, then only one set of SCL-SDA (J4) jumpers needs to be connected.

I²C address configuration can be done through UART TTL interface.

LED M1 IND (D3) and M2 IND (D4) function as stepper motor condition indicator (direction, condition, or pulse indicator).

3. SPC NEO STEPPER MOTOR 1.2A INTERFACE

SPC NEO STEPPER MOTOR 1.2A has UART TTL and I²C interfaces that can be used to receive commands or send data.

3.1. UART TTL INTERFACE

UART TTL communication parameters are as follows:

- 38400 bps
- 8 data bits
- 1 stop bit
- no parity bit
- no flow control

All commands sent through UART TTL interface begin with 1 byte data that contains **<command number>**, followed by (if needed) n-byte data command parameter.

If the command sent is a command requesting data from the SPC NEO STEPPER MOTOR 1.2A module, then SPC NEO STEPPER MOTOR 1.2A will send the data via TX TTL line.

A data parameter that has a range larger than 255 decimals (larger than 1 byte) will be sent in two steps. 1 byte MSB data is sent first and is followed by LSB data. For example: parameter <pulse delay> which has a range of 1 - 65535. If <pulse delay> has a value of 1500 then MSB byte will be 5 and LSB byte will be 220 ((5x256)+220=1500).

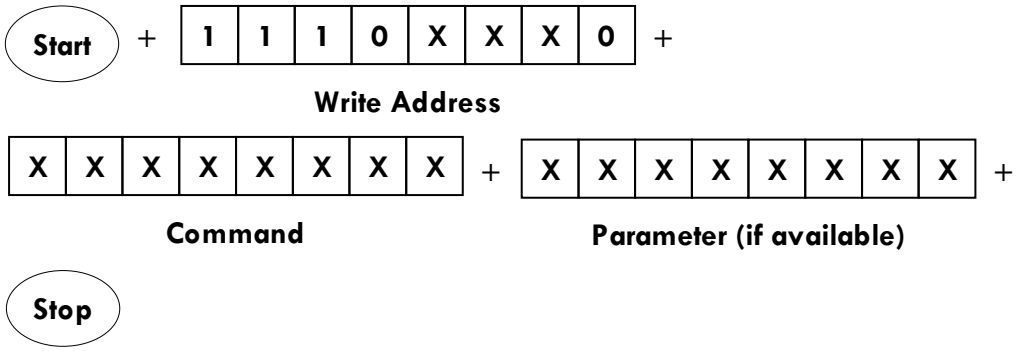
Available commands and parameters can be seen in **section 3.3**.

3.2. I²C INTERFACE

SPC NEO STEPPER MOTOR 1.2A module has a I²C interface. In this interface, SPC NEO STEPPER MOTOR 1.2A module acts as a slave with an address that has been determined via UART command (see **section 3.3.5**). I²C interface on SPC NEO STEPPER MOTOR 1.2A module supports bit rate up to a maximum rate of 50 kHz.

All commands sent through I²C interface begin with **start condition**, followed by 1 byte of SPC NEO STEPPER MOTOR 1.2A module address. After the address is sent, the master must send 1 byte data that contains **<command number>**, followed by (if needed) n-byte command parameter data. After all command parameters have been sent, the command is ended with **stop condition**.

The following is the sequence that must be done to send a command via I^2C interface.



A data parameter that has a range larger than 255 decimals (larger than 1 byte) will be sent in two steps. 1 byte MSB data is sent first and is followed by LSB data. For example: parameter <pulse delay> which has a range of 1 - 65535. If <pulse delay> has a value of 1500 then MSB byte will be 5 and LSB byte will be 220 ((5x256)+220=1500).

Available commands and parameters can be seen in **section 3.3**.

3.3. COMMAND SET

The following is a complete list of commands on UART and I^2C interface:

3.3.1. CONTINUOUS RUN

Function	Controls stepper motor so that it rotates continuously
Command	**0x30**
Parameter	<step type> 1 → Full-Step: motor will rotate 1 step every 1 pulse 2 → Half-Step: motor will rotate ½ step every 1 pulse <direction> 0 → motor will rotate clockwise 1 → motor will rotate counter clockwise <pulse delay> 1 - 65535 → Delay time between pulse to stepper motor. The smaller the pulse delay, the faster the stepper motor rotates
Response	-
Description	● If parameter <direction> has a value of 0, then M1 indicator LED will lit up red. ● If parameter <direction> has a value of 1, then M1 indicator LED will lit up green. ● If the stepper motor rotates to an opposite direction, then it means that the connection is reversed. To fix it, change the order of connection installation. ● Each time 1 step pulse is given, then M2 indicator LED will change from red to green or green to red (if the pulse delay is small enough, M2 indicator LED will lit up orange). ● One pulse delay value represents delay time between pulse for about 1 ms.

Example with UART interface to run the stepper motor so that it rotates clockwise continuously, with a full-step step type, and the delay between pulse is about 100 ms (0x0064 hexadecimal):

User : 0x30 0x01 0x00 0x00 0x64

The following is a pseudo code example, to use this command with I^2C interface (I^2C address example = 0xE0):

```
i2c_start();       // Start Condition
i2c_write(0xE0);   // Write to SPC Neo Stepper Motor module
i2c_write(0x30);   // "Continuous Run" command
i2c_write(0x01);   // Step type
i2c_write(0x00);   // Direction
i2c_write(0x00);   // MSB pulse delay
i2c_write(0x64);   // LSB pulse delay
i2c_stop();        // Stop Condition
```

3.3.2. PULSE COUNT RUN

Function	Controls stepper motor so that it rotates according to how many steps given
Command	**0x31**
Parameter	<step type> 1 → Full-Step: motor will rotate 1 step every 1 pulse 2 → Half-Step: motor will rotate ½ step every 1 pulse <direction> 0 → motor will rotate clockwise 1 → motor will rotate counter clockwise <pulse delay> 1 - 65535 → Delay time between pulse to stepper motor. The smaller the pulse delay, the faster the stepper motor rotates <pulse count> 1 - 65535 → the number of pulse sent to stepper motor
Response	-
Description	● If parameter <direction> has a value of 0, then M1 indicator LED will lit up red. ● If parameter <direction> has a value of 1, then M1 indicator LED will lit up green. ● If the stepper motor rotates to an opposite direction, then it means that the connection is reversed. To fix it, change the order of connection installation. ● Each time 1 step pulse is given, then M2 indicator LED will change from red to green or green to red (if the pulse delay is small enough, M2 indicator LED will lit up orange). ● One pulse delay value represents delay time between pulse for about 1 ms. ● After the number of pulse that has been released matches the pulse count, stepper motor will automatically stop (on brake condition) while still maintaining motor torque (current is still flowing through stepper motor coils).

Example with UART interface to run the stepper motor so that it rotates clockwise 20 pulses (0x0014 hexadecimal) with a full-step step type, and the delay between pulses is about 1000 ms (0x03E8 hexadecimal):

User : 0x31 0x01 0x00 0x03 0xE8 0x00 0x14

The following is a pseudo code example, to use this command with I^2C interface (I^2C address example = 0xE0):

```
i2c_start();       // Start Condition
i2c_write(0xE0);   // Write to SPC Neo Stepper Motor module
i2c_write(0x31);   // "Pulse Count Run" command
i2c_write(0x01);   // Step type
i2c_write(0x00);   // Direction
i2c_write(0x03);   // MSB pulse delay
i2c_write(0xE8);   // LSB pulse delay
i2c_write(0x00);   // MSB pulse count
i2c_write(0x14);   // LSB pulse count
i2c_stop();        // Stop Condition
```

3.3.3. BRAKE

Function	Stops the stepper motor while still maintaining motor torque (current is still flowing through stepper motor coils).
Command	**0x32**
Parameter	-
Response	-
Description	● This command can be given after the Continuous Run command. ● On brake condition, stepper motor will stop while still maintaining motor torque (current is still flowing through stepper motor coils). ● On brake condition, M1 and M2 indicator LEDs will lit up according to the last Run command.

Example with UART interface:

User : 0x32

The following is a pseudo code example, to use this command with I^2C interface (I^2C address example = 0xE0):

```
i2c_start();       // Start Condition
i2c_write(0xE0);   // Write to SPC Neo Stepper Motor module
i2c_write(0x32);   // "Brake" command
i2c_stop();        // Stop Condition
```

3.3.4. STOP

Function	Stops the stepper motor (current doesn't flow through stepper motor coils)
Command	**0x33**
Parameter	-
Response	-
Description	● This command can be given after Continuous Run, Pulse Count Run, or Brake command. ● On stop condition, stepper motor will stop and there will

	be no current flowing through the motor coils. ● On stop condition, M1 and M2 indicator LEDs lit up orange. ● Stop condition is the default condition when the SPC module is powered on.

Example with UART interface:

User : 0x33

The following is a pseudo code example, to use this command with I²C interface (I²C address example = 0xE0):

```
i2c_start();       // Start Condition
i2c_write(0xE0);   // Write to SPC Neo Stepper Motor module
i2c_write(0x33);   // "Stop" command
i2c_stop();        // Stop Condition
```

3.3.5. SET I²C ADDRESS

Function	Changes I²C address
Command	**0x41**
Parameter	<0xAA> <0x55> <newAddress>
Response	-
Description	● This command can only be performed via UART communication line. ● SPC module will use the new I²C address after going through power off sequence. ● The allowed I²C address <newAddress> can be seen in the table below. ● If the new address given is incorrect, then the I²C address will not be changed (the previous address will be used). ● The default I²C address is 0xE0. ● I²C address data will be saved in EEPROM so it won't be erased when it's powered off.

I²C Address	
I²C Write Address	**I²C Read Address**
0xE0	0xE1
0xE2	0xE3
0xE4	0xE5
0xE6	0xE7
0xE8	0xE9
0xEA	0xEB
0xEC	0xED
0xEE	0xEF

Example with UART interface to change the I²C address from 0xE0 to 0xE2:

User : 0x41 0xAA 0x55 0xE2

3.3.6. READ I²C ADDRESS

Function	Reads the current I²C address
Command	**0x42**
Parameter	-
Response	<I²CAddress>

Description	• This command can only be performed via UART communication line. • SPC module's I²C address can also be seen through the number of blinks of the indicator LED when the module is powered on. • If the I²C address is 0xE0 then the indicator LED will blink green once. If the I²C address is 0xE2 then the indicator LED will blink green twice. If the I²C address is 0xE4 then the indicator LED will blink green 3 times, and so on until I²C address 0xEE at which the indicator LED will blink green 8 times.

Example with UART interface:

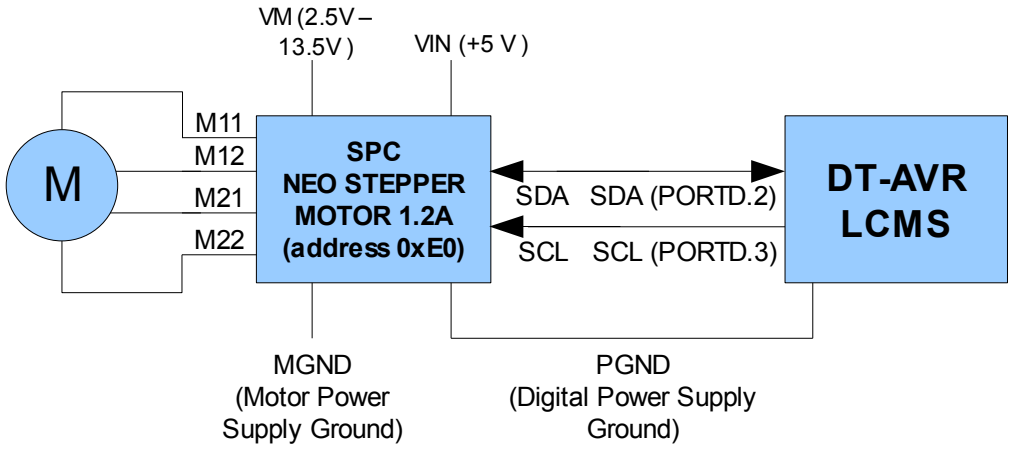
User : 0x42
SPC Module : <I2CAddress>

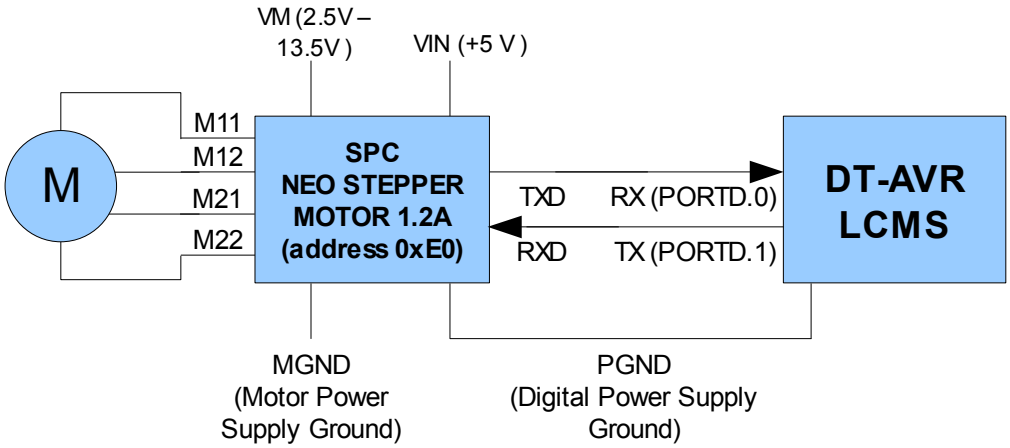
4. TESTING PROCEDURE

1. Connect the 5 Volts power supply to VIN and VM SPC NEO STEPPER MOTOR 1.2A module.
2. After the indicator LED blinks green according to I²C address, M1 and M2 indicator LEDs will lit up orange.
3. Send "Pulse Count Run" command as shown on the example in section 3.3.2 via UART TTL interface.
4. M1 indicator LED will lit up red and M2 indicator LED will change from red to green or green to red 10 times.
5. If the voltage is measured between pin M11 and M12 when M2 indicator LED is blinking, then the results will alternate between +VM and -VM.
6. If the voltage is measured between pin M21 and M22 when M2 indicator LED is blinking, then the results will alternate between +VM and -VM.

5. APPLICATION AND PROGRAM EXAMPLE

As an application example, SPC NEO STEPPER MOTOR 1.2A is used to run a motor stepper with I²C or UART interface. DT-AVR Low Cost Micro System (LCMS) module with ATmega8535 microcontroller is used as master that sends commands. The following are the connections between the modules:





As an example program for the above application, there are two programs named contoh_i2c.c and contoh_uart.c (included in the CD/DVD) written using CodeVisionAVR 1.25.2 evaluation.

In the program, DT-AVR LCMS will send "Continuous Run" command to SPC module (for example, SPC's I^2C address is 0xE0) so that the motor stepper rotates clockwise with 500 ms delay between pulses. After the command is sent, DT-AVR LCMS will wait for 5000 ms. Afterward "Brake" command will be sent to SPC followed by another 5000 ms delay. Then DT-AVR LCMS will send "Continuous Run" command to SPC module (for example, SPC's I^2C address is 0xE0) so that the motor stepper rotates counter clockwise for 10 pulses with 250 ms delay between pulses. This command is also followed by a 5000 ms delay. The program ends with DT-AVR LCMS sending "Stop" command to the SPC module.

ATTACHMENT A.
SPC NEO STEPPER MOTOR 1.2A Schematics